Express Mail Label No. EL875055455US

[0001] **TURBO DECODER WITH CIRCULAR REDUNDANCY CODE SIGNATURE COMPARISON**

[0002] CROSS-REFERENCE TO RELATED APPLICATION

[0003] This application claims priority from U.S. Provisional Patent Application Serial No. 60/248,440, filed November 14, 2000.

[0004] FIELD OF THE INVENTION

[0005] The present invention relates to communications systems which use error correcting for received communication signals and, in particular, to such systems which utilize iterative turbo decoder systems.

[0006] BACKGROUND

[0007] Turbo codes are a form of error correcting codes that yield performance near the Shannon limit for performance in an Additive While Gaussian Noise (AWGN) channel in a wireless communication system, such as time division duplex using code division multiple access (TDD/CDMA). Decoders for these codes utilize an iterative algorithm which gives an improved estimate of the transmitted data at each iteration.

[0008] A significant design parameter for decoders is the number of iterations to be used. Decoders can be implemented in hardware or software, but in either case the number of iterations used drives the requirement for processing resources, including the processing throughput required to achieve the desired data rate, power consumed in decoding, and the amount of hardware needed in a hardware implementation.

[0009] Two general strategies are known in the art for determining the number of iterations in a decoder implementation. First, a fixed number of iterations can be determined as part of the design. This simplifies the implementation, but requires excessive processing

resources since the fixed number must be set high enough to give the desired performance, i.e. bit error rate for the expected range of signal to noise levels, for nearly all cases where many decodings would require less than the fixed number of iterations.

[0010] Another strategy is to use a stopping rule to dynamically determine when decoding can be terminated without significantly effecting performance. The simplest stopping rule is the hard-decision-aided (HDA) criteria. When using this stopping rule, decoding is terminated when two successive iterations yield the same results. There are no changes in the hard decisions between iterations. Implementation of this rule for a coded block of N bits requires N memory locations to store the results of the previous implementation, as well as comparison of the previous N bit result to the current N bit result.

[0011] A typical turbo decoder may produce turbo decoder estimate data having in excess of 5,000 bits of information for each iteration. Accordingly, the implementation of a conventional stopping rule requires an additional memory allocation in excess of 5,000 bits to store a first code iteration for comparison with a next code iteration in order to determine whether the same results have been produced.

[0012] The inventor has recognized that it would be desirable to provide an improved turbo decoder which can more efficiently implement a stopping rule with a lesser requirement for additional memory.

[0013] SUMMARY

[0014] An iterative turbo decoder and method for error correcting communication signal data are provided. The decoder recursively evaluates signal data for a selected number of iterations.

[0015] During each iteration, decoder circuitry produces a new estimate of the transmitted data block, also called the extrinsics. A decoder data memory stores the extrinsics generated for one decoding iteration.

[0016] Signature code generating circuitry generates code signatures corresponding to each new estimate of the transmitted data block for each decoder iteration. The code signatures are preferably at least 20 times smaller than the data which they represent and for practical purposes will normally be at least 100 times smaller. A relatively small code signature memory stores the code signature corresponding to turbo decoder estimate data generated for one decoding iteration.

[0017] A comparator is operatively associated with the signature code circuitry and decoder circuitry. The comparator compares a generated code signature for a new estimate of the transmitted data block being produced and stored for a present decoder iteration with the contents of the signature memory. If the comparison reflects equality, the decoder circuitry ceases iteration processing. If the comparison reflects inequality, the generated code signature is stored in the signature memory where it is available for comparison relative to a code signature for a next decoder iteration.

[0018] The comparator may be used to store the generated code in the signature register. As an alternative, the comparator may simply access the signature register before the signature code generator outputs the new signature code. This permits the signature code generator to output the new signature code to both the comparator and to the signature register, as indicated in phantom, which eliminates the need for the comparator to perform a store operation to the signature code register.

[0019] Preferably, the comparator is operatively associated with the decoder circuitry to control decoder circuitry iteration processing only after a selected minimum number of iterations have occurred. Also, preferably the decoder circuitry ceases iteration processing if a predetermined limit of iterations has occurred. The limit of iterations is preferably an integer at least three greater than the selected minimum number. In a preferred embodiment, the selected minimum number is four (4) and the limit is eight (8).

[0020] It is an object of the present invention to provide an iterative turbo decoder which selectively implements a stopping rule with a lesser memory requirement than the prior art.

[0021] Other objects and advantages of the present invention will be apparent from the following description of a presently preferred embodiment.

[0022] BRIEF DESCRIPTION OF THE DRAWING(S)

[0023] Figure 1 is a schematic diagram of a turbo decoder made in accordance with the teachings of the present invention.

[0024] DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT(S)

[0025] With reference to Figure 1, there is shown a turbo decoder 10 having a communication signal input 12 and an output 14. The turbo decoder 10 includes turbo decoding iteration processing circuitry 20 and an associated turbo data register 22. The decoder processing circuitry 20 receives data blocks of communication signals via input 12 and generates a new estimate of the transmitted data block which is stored in register 22. The processing circuitry 20 is recursively associated with the turbo data register 22 such that the processor 20 utilizes the contents of the turbo data register 22 for the second and each successive iteration of turbo decoding processing.

[0026] The turbo decoding processing circuitry 20 is preferably configured with a predetermined limit as to the number of processing iterations which will occur for any given block of communication data such that the turbo decoder output is based upon the contents of the turbo decoder register after the last decoding iteration. Preferably, the maximum number of processing iterations performed by the processor 20 is eight (8).

[0027] The processor 20 also implements a stopping rule where fewer than the maximum number of iterations are needed. When the decoder determines that the estimate data being generated for successive iterations is not changing, iterative processing is stopped.

In lieu of providing a relatively large amount of additional memory to store a prior iteration of estimate data, a relatively simplistic signature code generator 24 and a relatively small code signature register 26 are provided as inputs to a comparator 28 which is operatively associated with the iteration processor 20 to implement the stopping rule.

[0028] Preferably, the comparator 28 is operatively associated with the decoder circuitry 20 to control decoder circuitry iteration processing only after a selected minimum number of iterations have occurred. Also, preferably the decoder circuitry 20 ceases iteration processing if a predetermined limit of iterations has occurred. The limit of iterations is preferably an integer at least three greater than the selected minimum number. In a preferred embodiment, the selected minimum number is four (4) and the limit is eight (8).

[0029] For a turbo decoder which generates binary estimate data on the order of 5,114 bits for a single iteration, the signature code generator preferably comprises a simple 16-bit binary divider which divides the 5,114 binary string of data by a selected 16-bit binary number and outputs the remainder which results from the division function to the comparator 28. The remainder will necessarily not exceed 16 bits since the divisor is 16 bits in length.

[0030] For a 16-bit divisor, preferably the binary number 1000000000000011 is utilized. Such a divisor corresponds to a binary polynomial represented as $1 + x^{14} + x^{15}$. The binary division performed by code generator 24, mathematically corresponds to dividing a binary polynomial representation of the 5,114 bit iteration estimate data by the polynomial $1 + x^{14} + x^{15}$ using binary (i.e. modulo 2) mathematics. The remainder of the binary division corresponds to the remainder polynomial. The odds that the remainder will be the same for two successive 5,114 bit string of estimate data are about 1 in 2^{16} which the inventor has determined is an acceptable risk factor.

[0031] Mathematical correspondence and use of polynomial representations to generate signal codes is known in the art and is discussed in, Pearson, W.W. and Brown, D.T., "Signal Codes For Error Detection", Proceedings of the IRE", January 1961. The inventor has recognized that this form of encoding has application to turbo decoders.

[0032] In operation, the turbo decoder processor 20 outputs, for a given iteration, N bits of estimate data to the turbo data register 22 and signal code generator 24. The signal code generator 24 generates a corresponding code signature having M bits which is preferably at least 100 times smaller than N which is input to the comparator 28. The comparator 28 compares the M bit signature code input from the code generator 24 with the contents of the signature register 26 to determine if they are equal.

[0033] If the comparator determines equality, a signal is sent to the processor 20 to stop iteration processing and output the turbo coding results. If the comparator detects inequality, the M-bit signature code received from the signature code generator 24 is stored in the signature register 26.

[0034] The comparator 28 may be used to store the generated code in the signature register 26. As an alternative, the comparator 28 may simply access the signature register 26 before the signature code generator 24 outputs the new signature code. This permits the signature code generator 24 to output the new signature code to both the comparator 28 and to the signature register 26, as indicated in phantom, which eliminates the need for the comparator 28 to perform a store operation to the signature code register 26.

[0035] Where a 5,114 bit block of binary data is produced for a decoder iteration, the signature code generator 24 preferably divides by 1000000000000011 to produce a remainder of no greater than 16 bits so that the signature register 26 need only have a 16-bit storage capacity.

[0036] The present invention is particularly suited to hardware implementations where the cost of generating the signature code is small, and the cost of the additional memory required would be high. It can also be used, however, in software implementations.

* * *